

Report and discuss

Untersuchung über die Lösung des Problems „pi“

Nachdem ich mir das PDF-Dokument angesehen hatte, begann ich mich dafür zu interessieren, wie man den Wert von Pi genauer berechnen kann. Zusätzlich zu Monte-Carlo-Algorithmen und numerischen Methoden begann ich nach anderen Möglichkeiten zur Berechnung von Pi zu suchen.

Dann sah ich, dass der antike griechische Mathematiker Archimedes die Methode der inneren und äußeren Polygone verwendete, um den Wert von Pi zu komprimieren.

Stellen wir uns einen Einheitskreis vor (also einen Kreis mit Radius 1).

Dann zeichnen wir ein in den Kreis eingeschriebenes regelmäßiges Polygon und ein in den Kreis umschriebenes regelmäßiges Polygon.

Wenn die Anzahl der Seiten eines Polygons zunimmt, nähern sich die Umfänge der beiden Polygone allmählich dem Umfang des Kreises an, der 2π beträgt.

Durch die Berechnung der Umfänge dieser beiden Polygone können wir eine untere und obere Grenze für π ermitteln.

Ich versuche diesen Code zu schreiben und es ist etwas schwierig. (in „andere Methode.cpp“)

```
After 1536 sides:
Lower bound: 3.14159
Upper bound: 1.75706

After 3072 sides:
Lower bound: 3.14159
Upper bound: 1.75706
```

Ich weiß nicht, warum der Wert der Untergrenze nicht auf ähnliche Weise an 3,14159 angenähert werden kann.

Obwohl das Endergebnis falsch war, war es sehr interessant. Ich habe auch nach dem Alter von Archimedes gesucht.

„Archimedes von Syrakus (griechisch Ἀρχιμήδης ὁ Συρακούσιος Archimédēs ho Syrakousios; * um 287 v. Chr. vermutlich in Syrakus; † 212 v. Chr. ebenda) war ein griechischer Mathematiker, Physiker und Ingenieur. Er gilt als einer der bedeutendsten Mathematiker der Antike. Seine Werke waren auch noch im 16. und 17. Jahrhundert bei der Entwicklung der höheren Analysis von Bedeutung.“ (aus Wiki)

Die Schlussfolgerung ist, dass dies ein sehr großartiger Wissenschaftler ist, der schon in so frühen Jahren eine so wirkungsvolle Methode entwickeln konnte, um das Problem der Pi-Genauigkeit zu lösen.

Diskussion über Computer System

Diskussion über "The Operating System Manages the Hardware"

1. Mein erster Eindruck von der Rolle des Betriebssystems

Das Betriebssystem dient als Schicht zwischen der Anwendungssoftware und der Hardware. Es schützt die Hardware vor Missbrauch durch fehlerhafte Anwendungen und bietet Anwendungen einfache und einheitliche Mechanismen zur Steuerung der Hardware. Dies wurde mir besonders klar, als ich las, wie das Betriebssystem Prozesse, virtuellen Speicher und Dateien als Abstraktionen bereitstellt.

2. Die Verbindung zu vorherigen Kapiteln

In den ersten Kapiteln wurde die Funktionsweise von Computern und die Rolle von Prozessoren erläutert. Es wurde betont, dass Prozessoren Anweisungen aus dem Speicher lesen und interpretieren. Mit diesem Wissen konnte ich besser verstehen, wie das Betriebssystem als Vermittler zwischen Anwendungen und Hardware fungiert.

Die Tatsache, dass das Betriebssystem als Schutzschild für die Hardware dient und gleichzeitig Anwendungen ermöglicht, die Hardware effizient zu nutzen, ist beeindruckend. Es zeigt, wie komplex und doch organisiert moderne Computersysteme sind. Besonders interessant fand ich die Abstraktionen, die das Betriebssystem bereitstellt, wie Prozesse, virtueller Speicher und Dateien. Diese Abstraktionen vereinfachen die Interaktion mit der Hardware und ermöglichen es Programmierern, effizientere Programme zu schreiben.

Diskussion über "Concurrency and Parallelism"

1. Mein Verständnis von Nebenläufigkeit und Parallelität

Beim Durchlesen des Kapitels wurde mir bewusst, dass Nebenläufigkeit und Parallelität zwei Konzepte sind, die eng miteinander verbunden sind, aber dennoch unterschiedliche Bedeutungen haben. Während Nebenläufigkeit sich auf das gleichzeitige Ausführen von mehreren Aktivitäten bezieht, zielt Parallelität darauf ab, durch diese Nebenläufigkeit die Systemleistung zu verbessern.

2. Die Verbindung zu vorherigen Kapiteln

In den ersten Kapiteln des Buches wurde die Funktionsweise von Computern und die Rolle von Prozessoren erläutert. Es wurde betont, dass Prozessoren Anweisungen aus dem Speicher lesen und interpretieren. Mit diesem Wissen konnte ich besser verstehen, wie Nebenläufigkeit

auf Prozessorebene funktioniert, insbesondere bei modernen Prozessoren, die mehrere Anweisungen gleichzeitig ausführen können.

Die Idee, dass ein Computer mehrere Dinge gleichzeitig tun kann, ist faszinierend. Es eröffnet Möglichkeiten für schnellere und effizientere Programme. Besonders beeindruckend fand ich den Abschnitt über Single-Instruction, Multiple-Data (SIMD) Parallelität. Die Tatsache, dass moderne Prozessoren spezielle Hardware haben, die es ermöglicht, mit einer einzigen Anweisung mehrere Operationen parallel auszuführen, zeigt das unglaubliche Potential und die Komplexität moderner Systeme.